

1 **NopeRoomGS: Indoor 3D Gaussian Splatting Opti-**
2 **mization without Camera Pose Input**

3
4 **— Technical Appendices and Supplementary Material**

5 **Contents**

6	A Preliminary of 3D gaussian splatting	2
7	B Camera Pose Optimization using Lie Algebra	2
8	C Local-to-Global Strategy	3
9	D Global objective function	3
10	E Additional Experiment Results	4
11	E.1 Dataset Details	4
12	E.2 Additional Comparison Results with COLMAP	4
13	E.3 Comparison with SLAM	5
14	E.4 Runtime and Memory Overhead Analysis	6
15	E.5 Additional Ablation Study	6
16	E.6 Additional Qualitative Results	6

17 A Preliminary of 3D gaussian splatting

18 3D Gaussian Splatting (3DGS) [9] represents a 3D scene as a set of anisotropic Gaussian primitives,
 19 each defined by a spatial mean $\boldsymbol{\mu} \in \mathbb{R}^3$ and a full covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3 \times 3}$. The spatial density
 20 of a Gaussian at location $\mathbf{x} \in \mathbb{R}^3$ is modeled as:

$$\mathcal{G}(\mathbf{x}) = \exp \left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right). \quad (1)$$

21 In addition to spatial geometry, each Gaussian is equipped with a set of view-dependent appearance
 22 parameters, including spherical harmonics (SH) coefficients for representing directional radiance, an
 23 opacity term α_i , and per-instance affine transformations that govern its scale and orientation.

24 To render a scene from a given camera viewpoint, each Gaussian is first transformed from world
 25 coordinates into the camera coordinate frame using a rigid-body transformation $\mathbf{T} = [\mathbf{R} \mid \mathbf{t}] \in$
 26 $SE(3)$, where $\mathbf{R} \in SO(3)$ denotes rotation and $\mathbf{t} \in \mathbb{R}^3$ denotes translation. The transformed mean
 27 $\mathbf{T}\boldsymbol{\mu}$ is then projected onto the image plane via a perspective projection function $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. Based
 28 on the Jacobian \mathbf{J} of the projective transformation, the 2D center of the Gaussian on the image plane,
 29 $\boldsymbol{\mu}^{2D}$ and its projected covariance $\boldsymbol{\Sigma}^{2D}$ are given by:

$$\boldsymbol{\mu}^{2D} = \pi(\mathbf{T}\boldsymbol{\mu}), \boldsymbol{\Sigma}^{2D} = \mathbf{J}\mathbf{R}\boldsymbol{\Sigma}\mathbf{R}^\top \mathbf{J}^\top. \quad (2)$$

30 To compute the final color of a pixel \mathbf{p} , The rendered color of a pixel, denoted by C_p can be calculated
 31 by alpha blending:

$$C_p = \sum_i c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

32 where c_i and α_i denote the color and opacity properties of the i -th projected Gaussian, respectively.
 33 A similar formulation is used to compute per-pixel depth by substituting c_i with the depth d_i of the
 34 Gaussian: $D_p = \sum_i d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)$.

35 B Camera Pose Optimization using Lie Algebra

36 Pose-free reconstruction demands accurate and efficient gradient computation with respect to the
 37 camera pose. However, existing rasterization-based differentiable renderers, as described in Sec. A,
 38 fail to provide such gradients due to their black-box treatment of the projection process. To address
 39 this gap, we analytically derive the gradients of projected Gaussian parameters with respect to the
 40 camera pose $\mathbf{T}_i \in SE(3)$ (the subscript i is omitted in this subsection for clarity) using Lie algebra
 41 tools, enabling fully differentiable optimization over $SE(3)$ transformations.

42 By applying the chain rule and based on Eq. 2, the gradient of projected Gaussian parameters with
 43 respect to camera pose $\mathbf{T} = [\mathbf{R} \mid \mathbf{t}]$ can be expressed as:

$$\frac{\partial \boldsymbol{\mu}^{2D}}{\partial \mathbf{T}} = \frac{\partial \boldsymbol{\mu}^{2D}}{\partial \boldsymbol{\mu}^C} \frac{\partial \boldsymbol{\mu}^C}{\partial \mathbf{T}}, \quad (4)$$

$$\frac{\partial \boldsymbol{\Sigma}^{2D}}{\partial \mathbf{T}} = \frac{\partial \boldsymbol{\Sigma}^{2D}}{\partial \mathbf{J}} \frac{\partial \mathbf{J}}{\partial \boldsymbol{\mu}^C} \frac{\partial \boldsymbol{\mu}^C}{\partial \mathbf{T}} + \frac{\partial \boldsymbol{\Sigma}^{2D}}{\partial \mathbf{R}} \frac{\partial \mathbf{R}}{\partial \mathbf{T}}. \quad (5)$$

44 To compute $\partial \boldsymbol{\mu}^C / \partial \mathbf{T}$ analytically, we leverage the Lie group structure of $SE(3)$. Specifically, a small
 45 perturbation $\delta \xi \in \mathfrak{se}(3)$ in the tangent space induces a left-multiplied transformation on the original
 46 pose \mathbf{T} . The partial derivative on the manifold is defined as:

$$\frac{\mathcal{D}f(\mathbf{T})}{\mathcal{D}\mathbf{T}} \triangleq \lim_{\tau \rightarrow 0} \frac{\log(f(\exp(\tau) \circ \mathbf{T}) \circ f(\mathbf{T})^{-1})}{\tau}, \quad (6)$$

47 where \circ denotes group composition, and \exp, \log are the logarithmic and exponential maps between
 48 Lie Algebra and Lie Group. With this, we derive the following:

$$\frac{\mathcal{D}\boldsymbol{\mu}^C}{\mathcal{D}\mathbf{T}} = [I \quad -\boldsymbol{\mu}_C^\times], \quad \frac{\mathcal{D}\mathbf{R}}{\mathcal{D}\mathbf{T}} = \begin{bmatrix} 0 & -\mathbf{R}_{:,1}^\times \\ 0 & -\mathbf{R}_{:,2}^\times \\ 0 & -\mathbf{R}_{:,3}^\times \end{bmatrix}, \quad (7)$$

where \times denotes the skew-symmetric matrix of a 3D vector, and $\mathbf{R}_{:,i}$ is the i -th column of the projected basis matrix. These derivatives enable end-to-end differentiability of our renderer with respect to pose, which is critical for stable and efficient optimization in pose-free 3D Gaussian Splatting.

C Local-to-Global Strategy

In low-texture scenarios, traditional geometric methods often struggle due to a lack of reliable visual features. Our method addresses this by integrating dense depth and correspondence priors from pretrained foundation models (Depth Estimation Foundation Model [8] and CoTracker [7]) during the local stage. These priors provide global structure-aware cues that are especially beneficial in regions where SIFT/SfM-based methods may fail. The priors act as soft constraints that guide the local optimization even in the absence of strong appearance gradients.

One of the key distinctions of our framework compared to prior pipelines (e.g., CF-3DGS [4]) lies in how initial camera poses are optimized. Specifically, we segment the video into short, overlapping clips and perform joint pose and geometry optimization at the clip level, rather than frame-by-frame. This clip-wise formulation improves robustness to fast camera motion in two ways:

- (1). Better-conditioned optimization. Each clip forms a small-scale multi-view reconstruction problem with overlapping views, which improves numerical stability and optimization convergence.
- (2). Stronger geometric constraints. Multi-view consistency across multiple frames enables more accurate pose disambiguation than pairwise (frame-to-frame) approaches, particularly when motion blur or abrupt motion causes image misalignment.

D Global objective function

To achieve accurate and stable reconstruction in pose-free settings, we design a global optimization objective that jointly refines the 3D Gaussian field \mathcal{G} and the camera poses \mathbf{T}_i across all input views. This global stage builds upon the local initialization and enforces multi-view consistency by jointly aligning both photometric appearance and geometric structure. Specifically, we integrate three complementary loss terms: a photometric loss that ensures view-consistent appearance, a depth alignment loss that anchors geometry using locally estimated depth maps, and a pose regularization term based on a piecewise planarity assumption, which enhances robustness under textureless regions and abrupt camera motion.

Photometric loss. Following the original 3DGS [9], we include the photometric loss terms in the objective function:

$$\mathcal{L}_{\text{rgb}}^i = \gamma \|I_{\mathcal{G}}(\mathbf{T}_i) - I_i\|_1 + (1 - \gamma) \mathcal{L}_{D-SSIM}(I_{\mathcal{G}}(\mathbf{T}_i), I_i), \quad (8)$$

which is constituted by a L_1 term and a D-SSIM term [9]. $I_{\mathcal{G}}(\mathbf{T}_i)$ represents the rendered image from the Gaussian Splatting \mathcal{G} with the camera pose \mathbf{T}_i , I_i is the i -th input image, and γ is the hyperparameter to balance the two terms. In our experiments, we set the parameter $\gamma = 0.2$.

Depth alignment loss. With the photometric loss, the camera poses can be effectively corrected by the regions with rich textures, as these regions can produce strong gradients if the poses are erroneous. However, the indoor scenes are full of textureless areas, e.g., blank walls, which pose great challenges to pose-free 3DGS. To enhance the robustness in textureless regions, we introduce a depth-based constraint:

$$\mathcal{L}_{\text{depth}}^i = \rho \left(D_{\mathcal{G}}(\mathbf{T}_i) - \alpha D_{\theta,i} - \beta \right). \quad (9)$$

The difference is that we here use the optimized depth maps $D_{\theta,i}$ from the local stage as the ground truth to constrain the depth maps $D_{\mathcal{G}}(\mathbf{T}_i)$ rendered with camera pose \mathbf{T}_i .

Pose constraint based on piecewise planarity assumption. The camera motion could be abrupt in indoor scenes. To further improve the robustness against abrupt camera motion, we propose a cross-frame geometric constraint based on a piecewise planarity assumption.

We assume each point \mathbf{x}_p in the scene lies on a infinitesimal piecewise plane defined by $\mathbf{n}_i^\top \mathbf{x}_p + \delta_i = 0$, where \mathbf{n}_i is the surface normal and δ_i is the displacement coefficient. The normal and the displacement coefficient are calculated from the rendered depth map of 3DGS \mathcal{G} , using 4 surrounding pixels (left, right, upper, lower).

Given two consecutive frames, I_i and I_{i+1} with a relative pose estimation $\mathbf{T}_{i \rightarrow i+1} = [\mathbf{R}_{i \rightarrow i+1} \mid \mathbf{t}_{i \rightarrow i+1}]$ between, the plane parameters in frame i , denoted as (\mathbf{n}_i, δ_i) , can be transformed to frame

99 $i + 1$ by

$$\hat{\mathbf{n}}_{i+1} = \mathbf{R}_{i \rightarrow i+1} \mathbf{n}_i, \quad (10)$$

$$\hat{\delta}_{i+1} = \delta_t - \mathbf{n}_i^\top \mathbf{t}_{i \rightarrow i+1}, \quad (11)$$

100 to get the transformed plane parameters $(\hat{\mathbf{n}}_{i+1}, \hat{\delta}_{i+1})$.

101 We then enforce consistency between the transformed plane parameters $(\hat{\mathbf{n}}_{i+1}, \hat{\delta}_{i+1})$ and the directly
102 estimated values $(\mathbf{n}_{i+1}, \delta_{i+1})$ in the frame $i + 1$, using the following loss function:

$$\mathcal{L}_{\text{plane}}^{i+1} = \lambda_n \|1 - \hat{\mathbf{n}}_{i+1}^\top \mathbf{n}_{i+1}\|_2^2 + \lambda_\delta \|\hat{\delta}_{i+1} - \delta_{i+1}\|_2^2, \quad (12)$$

103 where λ_n and λ_δ are the regularization parameters that control the relative importance of normal and
104 offset consistency, respectively. This loss function ensures that the geometries in consecutive frames
105 are coherently aligned and decomposes the supervision to rotational and translational components of
106 the camera poses. We found that, by adjusting λ_n and λ_δ , we can achieve more robust pose estimation
107 in indoor environments. To further enhance robustness, we also extract the edge map based on input
108 frame, which is used as a mask to restrict this constraint only to planar regions, while avoiding its
109 effect on the plane edges. Further detail can be found in the supplementary materials.

110 This loss function encourages consistent geometric representation
111 across adjacent frames by enforcing the alignment of plane param-
112 eters under relative pose transformations. Specifically, the formula-
113 tion decouples the supervision into two interpretable components:
114 normal direction alignment, associated with camera rotation, and dis-
115 placement consistency, linked to camera translation. This separation
116 facilitates stable gradient flow and enables more precise optimization
117 of pose parameters. Fig. 1 presents the Absolute Trajectory Error
118 (ATE) under varying combinations of the regularization param-
119 eters, λ_n and λ_δ , which modulate the relative influence of normal
120 and offset consistency in the proposed piecewise planar assumption.
121 Each grid cell corresponds to a specific $(\lambda_n, \lambda_\delta)$ configuration, with
122 darker colors indicating lower ATE. The results demonstrate that
123 excessively low weights lead to insufficient geometric regularization,
124 whereas overly high weights may impose excessive rigidity, hin-
125 dering accurate pose recovery. An appropriate setting consistently
126 yields minimal ATE, highlighting the importance of balanced regularization in achieving robust and
127 geometrically coherent camera pose estimation across scenes.

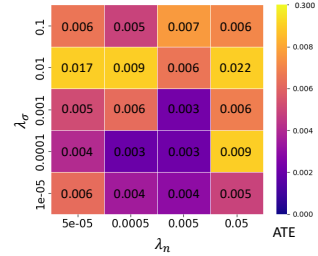


Figure 1: Effect of regularization parameters, λ_n and λ_δ , on Absolute Trajectory Error (ATE) under piecewise planarity assumption.

128 E Additional Experiment Results

129 E.1 Dataset Details

130 We evaluate our method on three public datasets: Replica [16], ScanNet [3], and Tanks & Tem-
131 ples [10], covering both synthetic and real-world scenarios. Replica [16] dataset is a high-fidelity
132 synthetic dataset featuring indoor environments with accurate ground-truth camera poses. It includes
133 large textureless regions and complex trajectories, making it ideal for evaluating pose estimation
134 and view synthesis under challenging conditions. To simulate realistic sparse-view reconstruction,
135 we uniformly sample every 10th frame, resulting in 80 training frames and 15 testing frames per
136 scene. This setup ensures sufficient scene coverage while introducing pose estimation challenges
137 due to large inter-frame motion and textureless regions. ScanNet [3] comprises real-world RGB-D
138 indoor sequences captured in unconstrained settings, introducing challenges such as motion blur,
139 occlusion, and sensor noise. Following prior work [5], we use the provided RGB frames. For each
140 sequence, we uniformly sample every 3th frame, resulting in 80 training frames and 15 testing frames
141 per scene. Tanks & Temples [10] dataset contains outdoor and indoor scenes with rich textures and
142 relatively stable camera motion. As a benchmark frequently used by prior work, it helps assess the
143 generalizability of our method under standard evaluation protocols.

144 E.2 Additional Comparison Results with COLMAP

145 **Replica.** We evaluate our method on Replica [16] dataset, which provides high-quality ground-truth
146 camera poses and dense geometry in synthetic indoor environments. To enable a more compre-
147 hensive and representative comparison for camera pose estimation and novel view synthesis, we

apply COLMAP [15] to the same set of input images using default Structure-from-Motion (SfM) configurations. This allows us to evaluate our method under fair conditions with access to the same image content and initialization constraints. The quantitative results are summarized in Tab. 1, where we report camera pose estimation metrics (RPE_t , RPE_r , ATE) and rendering quality metrics (PSNR, SSIM, LPIPS). Our method achieves consistently superior performance across all metrics. Notably, COLMAP [15] successfully recovers camera poses for only 5 out of 8 sequences; hence, its average results are computed over the successful cases only. In contrast, our method remains robust across all sequences, demonstrating high accuracy in both geometric alignment and visual quality.

Tanks & Temples. To assess generalizability beyond indoor domains, we evaluate our method on selected scenes from the Tanks & Temples benchmark [10], which features large-scale outdoor and indoor environments with rich textures and high-resolution imagery. For comparison, we include results from COLMAP [15], which is widely regarded as a strong classical baseline for SfM. We report both camera pose accuracy and novel view synthesis quality in Tab. 2, demonstrating our method’s ability to generalize across domains with varying levels of texture, geometry complexity, and motion smoothness.

E.3 Comparison with SLAM

Although NoPeRoomGS is designed as a pose-free 3D Gaussian reconstruction framework, it shares conceptual similarities with GS-based SLAM systems, as both aim to jointly recover camera trajectories and scene representations. However, the design philosophy and optimization paradigm differ substantially.

(1). Conceptual Connection and Distinctions

Conventional GS-based SLAM methods decompose the pipeline into two stages: a frame-by-frame tracking stage that estimates incremental camera poses, and a global optimization stage for loop closure and map refinement. Our framework can be viewed as an evolution of this idea—extending SLAM’s geometric reasoning into a pose-free, local-to-global optimization scheme. Specifically, our local stage leverages LNRR to jointly optimize multiple adjacent frames in a batch, improving robustness against tracking failure under low-texture or abrupt-motion conditions. The global stage then aggregates all local results to form a coherent scene representation, eliminating the need for explicit tracking or loop closure commonly used in SLAM.

Unlike SLAM systems that assume high frame rates and temporal continuity, NoPeRoomGS is tailored for offline novel view synthesis from unordered or sparsely captured image collections. This allows it to operate effectively even in scenes with discontinuous viewpoints or large inter-frame motion, where conventional SLAM often fails to initialize or maintain stable tracking.

(2). Quantitative Comparison

To quantitatively assess this distinction, we compare NoPeRoomGS with representative GS-based SLAM methods, as summarized in Tab. 3. Under identical experimental settings using RGB inputs, our method achieves superior camera pose accuracy and rendering fidelity, demonstrating that a pose-free, batched optimization strategy can rival, and even surpass, traditional tracking-based pipelines.

Table 1: Camera pose estimation and novel view synthesis performance metrics comparison on Replica [16] dataset. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol. The best results are highlighted in bold.

Methods	$RPE_t \downarrow$	$RPE_r \downarrow$	ATE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
COLMAP [14]+3DGS [9]	0.718	0.633	0.019	30.21	0.87	0.12
NeRFmm [17]	1.523	6.962	0.048	24.24	0.61	0.53
BARF [1]	1.230	6.051	0.046	25.86	0.65	0.46
Nope-NeRF [2]	1.141	6.001	0.039	27.46	0.75	0.40
SelfSplat [6]	1.531	7.560	0.072	18.90	0.54	0.42
NoPoSplat [18]	1.773	7.086	0.061	19.39	0.52	0.41
CF-3DGS [4]	1.150	5.903	0.059	25.40	0.77	0.29
Ours	0.148	0.126	0.009	32.14	0.90	0.09

Table 2: Camera pose estimation and novel view synthesis performance metrics comparison on Tanks & Temples [10] dataset. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol. The best results are highlighted in bold.

Methods	$RPE_t \downarrow$	$RPE_r \downarrow$	ATE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
COLMAP [14]+3DGS [9]	-	-	-	30.20	0.92	0.10
NeRFmm [17]	1.735	0.477	0.123	22.50	0.59	0.54
BARF [1]	1.046	0.441	0.078	23.42	0.61	0.54
Nope-NeRF [2]	0.080	0.038	0.006	26.34	0.74	0.39
SelfSplat [6]	1.046	0.489	0.094	22.42	0.58	0.56
NoPoSplat [18]	1.832	0.488	0.117	20.15	0.53	0.47
CF-3DGS [4]	0.041	0.069	0.004	31.28	0.93	0.09
Ours	0.034	0.043	0.003	31.68	0.94	0.07

Table 3: Comparison of methods in terms of camera pose accuracy and rendering quality on Replica [16] dataset.

Method	$RPE_t \downarrow$	$RPE_r \downarrow$	ATE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
GS3LAM [11]	0.933	2.354	0.043	24.35	0.69	0.36
SGS-SLAM [12]	0.732	3.322	0.037	25.45	0.70	0.33
MonoGS [13]	0.832	1.234	0.039	27.40	0.73	0.29
Ours	0.148	0.126	0.009	32.14	0.90	0.09

E.4 Runtime and Memory Overhead Analysis

To assess the computational efficiency of NoPeRoomGS, we compare its runtime and memory overhead with representative pose-free reconstruction frameworks, including Nope-NeRF [2] and CF-3DGS [4].

While Nope-NeRF [2] eliminates pre-computed poses via NeRF-based optimization, it remains computationally intensive due to per-frame neural field updates. CF-3DGS [4] improves efficiency through Gaussian parameterization but still requires incremental optimization with estimated poses. In contrast, our method unifies geometry and pose estimation in a single local-to-global optimization framework, achieving a balance between reconstruction quality and efficiency.

Our full pipeline takes approximately 81 minutes per scene on an NVIDIA RTX 4090D GPU, with peak memory usage under 12GB. The local stage introduces an additional 10-minute overhead (due to LNGR), while the global stage takes about 71 minutes.

As shown in Tab. 4, under the same 80-frame setting, our method achieves a favorable trade-off between runtime, memory usage, and reconstruction robustness. Although slightly slower than CF-3DGS [4], it provides greater stability in challenging indoor scenes and remains significantly more efficient and scalable than Nope-NeRF [2] in both training and rendering stages.

Table 4: Comparison of methods in terms of overhead.

Method	Time	Memory
NopeNeRF [2]	~ 400 min	6GB
CF-3DGS [4]	~ 63 min	9GB
Ours	~ 81 min	12GB

E.5 Additional Ablation Study

To further support and strengthen our experimental results, we conduct a series of ablation studies targeting four core components of our pipeline: (1) the local neural geometric representation (LNGR), (2) the piecewise planarity assumption (PPA), (3) the alternating optimization strategy (AOS), and (4) the depth alignment loss (DAL). Each component is individually removed or replaced with a simpler baseline to assess its impact on camera pose estimation and novel view synthesis, as shown in Tab. 5.

LNGR provides a shared depth representation across all frames in a clip, which encourages multi-view geometric consistency and helps reduce scale drift (a common issue in frame-wise optimization pipelines like CF-3DGS [4]). Such global coherence is difficult to maintain when depths are independently optimized for each frame.

To analyze LNGR’s contribution, we examine two baseline variants: The variant of "w/o LNGR v1" in Tab. 5, we replace LNGR with naive pose propagation and unrefined monocular depth, where the depth and pose of each current frame are initialized from the preceding frame; The variant of "w/o LNGR v2" in Tab. 5, we directly optimize per-pixel depth maps initialized from the same pretrained depth estimation network [8], allowing pixelwise refinement without enforcing cross-frame consistency.

Both variants lead to significant degradation in pose accuracy and rendering fidelity, as shown in Tab. 5, confirming the importance of LNGR for stable geometry learning and consistent reconstruction.

Eliminating PPA and using only pairwise depth consistency weakens global supervision, especially in textureless regions, confirming the geometric regularization effect of PPA. Replacing AOS with joint optimization hampers convergence, validating the need for decoupled updates to avoid interference between pose and geometry learning. Finally, discarding DAL and relying solely on raw monocular predictions for depth supervision leads to lower performance, demonstrating the value of using locally refined depth as a guiding signal in global optimization. Together, these results confirm that each module is essential for achieving stable, accurate, and photorealistic reconstruction outcomes.

Table 5: Ablation study on Replica [16] dataset. A comparison of our full pipeline and variants without Local Neural Geometric Representation (LNGR), Piecewise Planarity Assumption (PPA), Alternating Optimization Strategy (AOS), and Depth Alignment Loss (DAL) (described in Eq. 9), respectively.

Methods	ATE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
ours	0.009	32.14	0.90	0.09
w/o LNGR v1	0.063	14.82	0.65	0.42
w/o LNGR v2	0.049	28.46	0.82	0.34
w/o PPA	0.023	30.34	0.89	0.14
w/o AOS	0.019	28.28	0.89	0.15
w/o DAL	0.078	31.42	0.90	0.14

E.6 Additional Qualitative Results

To further demonstrate the effectiveness of our method in both camera pose estimation and novel view synthesis, we present additional qualitative comparisons in Fig. 2 and Fig. 3. As shown in Fig. 2, our method produces more accurate and consistent camera trajectory estimations (left) compared to

CF-3DGS [4], particularly in scenes that exhibit challenging conditions such as textureless walls and abrupt viewpoint changes. On the right, the corresponding synthesized views illustrate the superior photorealism and structural coherence of our approach. Fig. 3 provides a broader comparison against NoPe-NeRF [2] and CF-3DGS [4] across diverse indoor scenes. Our method consistently delivers higher-quality novel view synthesis, closely matching the ground truth (GT) even in low-texture and geometrically complex regions. These results validate the robustness of our method.

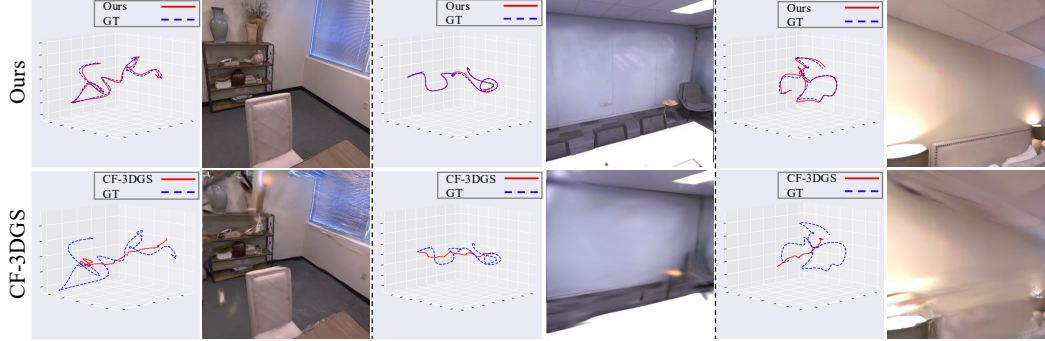


Figure 2: Comparison of camera pose estimation and novel view synthesis with the state-of-the-art. Compared to CF-3DGS [4] (bottom), our method (top) achieves more robust pose estimation and more photorealistic novel view synthesis in challenging indoor scenes with textureless regions or abrupt camera motion. Each example contains the camera trajectory estimation (left) and a sampled synthesized view (right).

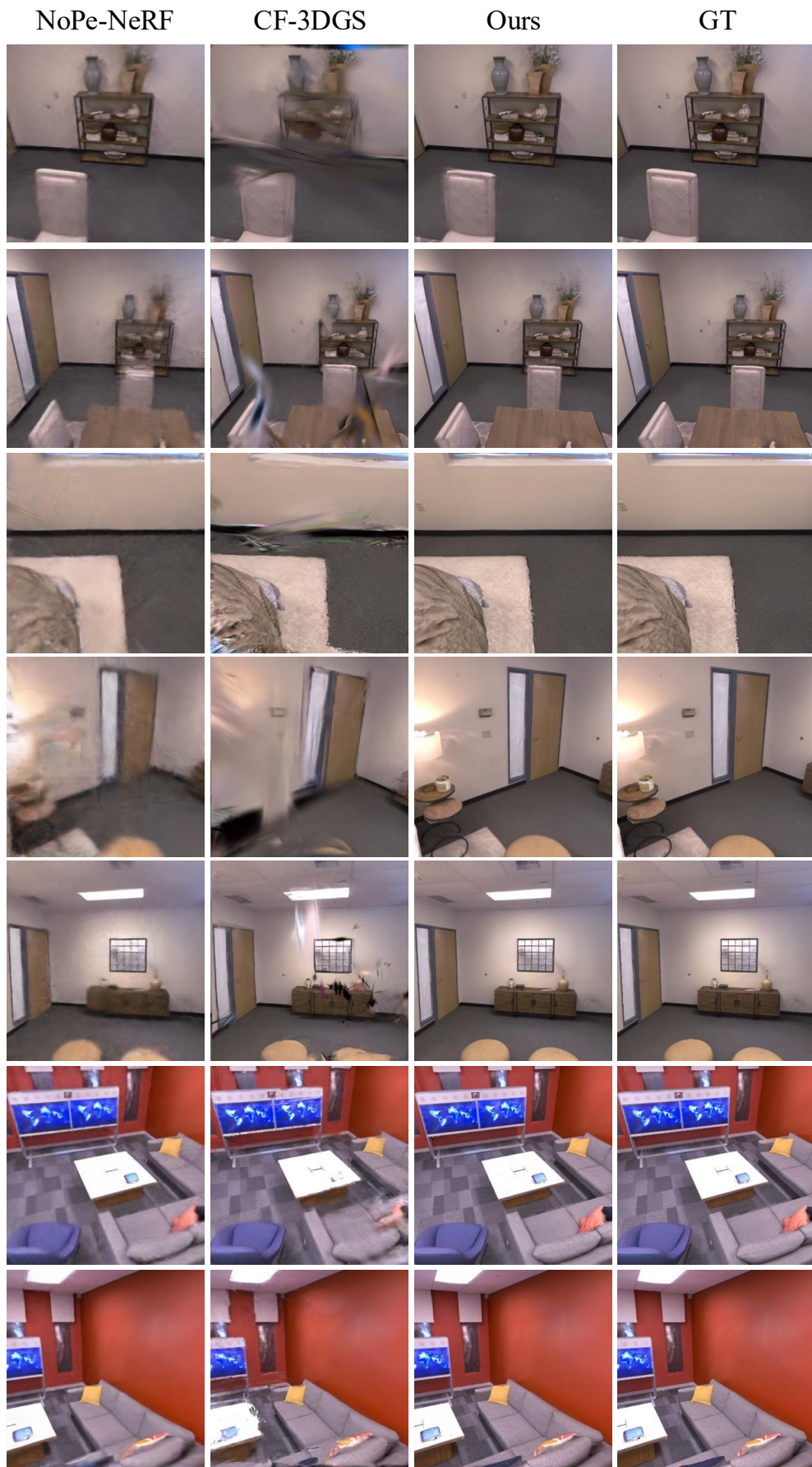


Figure 3: Qualitative comparison of novel view synthesis with state-of-the-art methods. Our method demonstrates superior robustness in pose estimation and photorealistic rendering quality, particularly in challenging indoor environments with textureless regions and abrupt camera motion.

References

- [1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.
- [2] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023.
- [3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.
- [4] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20796–20805, June 2024.
- [5] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5511–5520, 2022.
- [6] Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. *arXiv preprint arXiv:2411.17190*, 2024.
- [7] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pages 18–35. Springer, 2024.
- [8] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.
- [9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [10] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.
- [11] Linfei Li, Lin Zhang, Zhong Wang, and Ying Shen. Gs3lam: Gaussian semantic splatting slam. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3019–3027, 2024.
- [12] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024.
- [13] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024.
- [14] Johannes L Schönberger. *Robust methods for accurate and efficient 3D modeling from unstructured imagery*. PhD thesis, ETH Zurich, 2018.
- [15] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.
- [16] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

- 314 [17] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF—:
315 Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*,
316 2021.
- 317 [18] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou
318 Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images.
319 *arXiv preprint arXiv:2410.24207*, 2024.